Journal of Universal Computer Science, vol. 8, no. 4 (2002), 409-422
submitted: 10/8/01, accepted: 28/2/02, appeared: 28/4/02 © *J.UCS*

HWOES: A Hyperwave Online Employment Service

J. Lennon
(Department of Computer Science, University of Auckland
New Zealand
j.lennon@auckland.ac.nz)

H. Liu
(Department of Computer Science, University of Auckland
New Zealand
lhdh_h@hotmail.com)

H. Maurer
(Technical University and KNOW Center,
Graz, Austria
hmaurer@iicm.tu-graz.ac.at)

Abstract: In this paper we propose several significant advances in online employment services. We address issues such as privacy, interaction, and scalability to worldwide services. The details of each user are managed in completely relevant-to-service formats and both jobseekers & employers have considerable control over just how many of their own personal details are visible at any time. Perhaps most interestingly, virtual connections are maintained that support the various stages of negotiation in iterative, computer-supported cycles.

Keywords: e-commerce, online employment services, interactive negotiations
Category: J

1 Introduction

This paper investigates how Online Employment Services (OESs) can be designed to be significantly more interactive, personalized and scalable, while supporting user-controlled degrees of privacy. Certainly, OESs such as Monster [Mnstr 2001], HotJobs [HotJ 2001], Career Mosaic [HeadH 2001] and The Jobs Company [Jobs 2001] (to name but a few) handle millions of employment cases annually and are expanding their services steadily. However, interestingly, online negotiations between jobseekers and employers can now, we believe, be supported in ways that more closely resemble traditional methods used by conventional non-web employment services. These services aim to find the "right job for the right person" and it usually is an iterative process. We propose an application, supported by a knowledge-management system [Borghoff and Pareschi 1997], which generates in real time, transparency-controlled networks of virtual connections between jobseekers and employers.

The effectiveness of the system we propose is based on four factors:

(i) User-determined levels of privacy (see Section Two).

(ii) Iterative, computer-supported interactions between jobseekers and employers (see Section Three).

(iii) Scalability to worldwide services (see Section Five).

2 Transparency-Controllable Virtual Connections

Privacy is obviously a key issue. People may have confidence in a human agent but do they trust in an online system with just a few lines of privacy policy statement? A statement of privacy may not be good enough if users can just post their background materials without any options to "hide" something and control them [CntPub 2000]. Neither does it make them feel "safe" enough, if a sufficiently interactive mechanism is not provided. Furthermore, OESs should allow customers to request services like 'Please let me know what you have found for me before you show my real name (or other details) to the other side …'. Thus it should be the users themselves who are permitted to set which, and when, private details are visible. Hence, the virtual connections should be *transparency-controllable*. By this we mean that the system should provide interactive means to control various degrees of privacy for:

(i) Jobseekers/employers details. All users should be provided with convenient-to-use transparency settings.

(ii) Negotiation sessions. This is particularly important before users make their final decisions and reflects the levels of privacy found in employment services in the real world.

(iii) System administrators. System administrators can decide how many details both-sides can view and update. They will do this on the basis of options such as: per account, servicing session, period of time, etc.

All users are encouraged to offer more details about their background and wishes, but they can keep many details hidden in the first stages. It is important to note that searching and matching can be performed on all stored data and are not restricted to the visible attributes – even though users may be unable to see this information.

3 Negotiation Cycles Between Jobseekers and Employers

As mentioned previously, the matching of jobseekers and employers is usually achieved by negotiation. This means that no matter how good the matching algorithms used are, the system must support iterative cycles. For example, an employer may have set their salary range too low, with the expectation that they will raise it if there is a candidate that fulfills other attributes. Similarly, a jobseeker may not be experienced enough to even set a reasonable salary. We propose a system that alerts users at login time of any likely new candidates. Then, if any of these

candidates are of interest, the system can set up virtual connections. Candidates will be alerted and respond if interested. The process will be cyclic as shown in Figure 1.

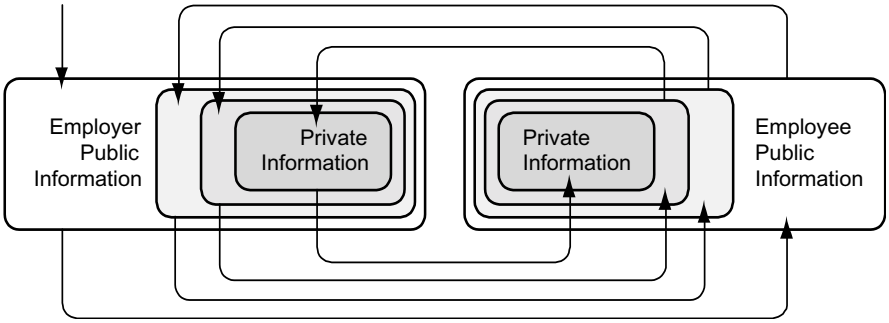

Figure 1: The Negotiation Cycle

Since all matching is done in real time we will avoid out-of-date messages such as those given by CGI scripts and email.

In the iterative negotiation cycle described above, we would expect to see transparency increase. This is because jobseekers can invoke the interactive functions of the system to request employers to reveal details appropriate to the stage of negotiation. And, of course, the converse is also true.

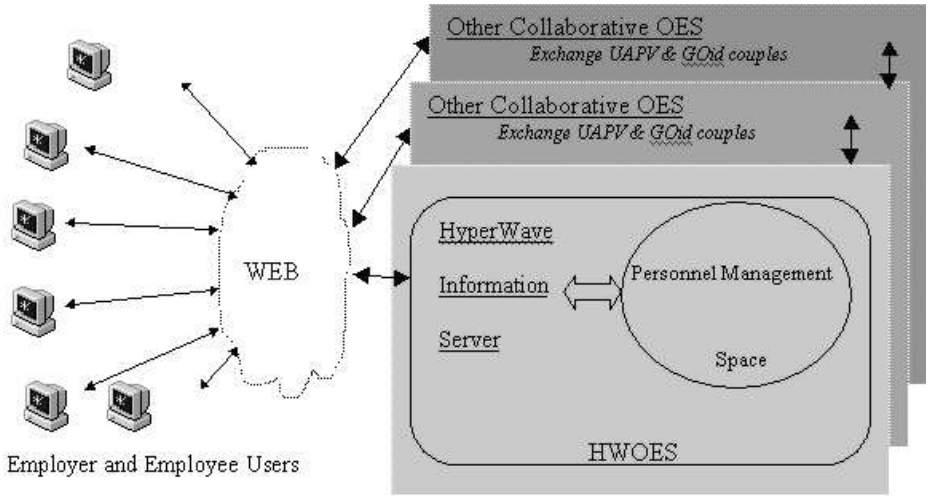

Figure 2: An Overview of the Hyperwave Online Employment System [Liu 2001]

4 The Hyperwave Online Employment System (HWOES).

The system we have implemented is based on Hyperwave [HW 2001; HWTW 1999; Maurer 1996] and is called the Hyperwave Online Employment System (HWOES). The implementation details for our prototype are described in [Liu 2001].
There are three main parts that form the environment for our online employment service (see Figure 2):

1. Users with Internet-accessible computers
2. The Web
3. Collaborative HWOES entities for international employment services. Users can use ordinary browsers like Microsoft IE or Netscape Navigator to access HWOES.

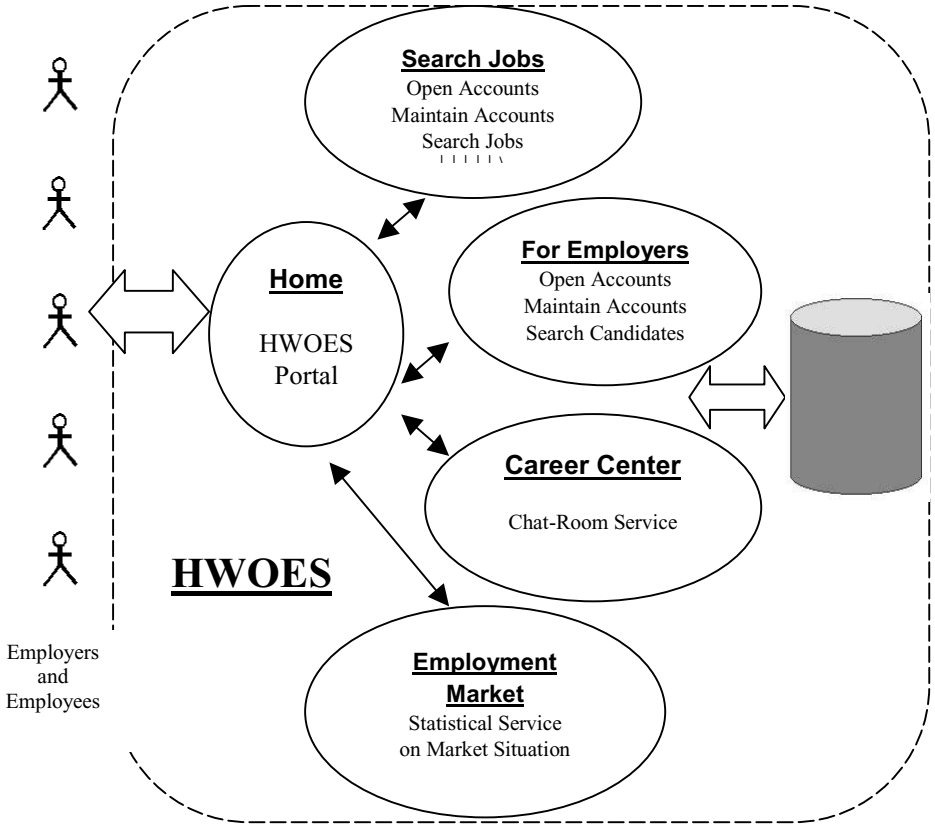

Figure 3: Main Functional Componenents of HWOES [Liu 2001]

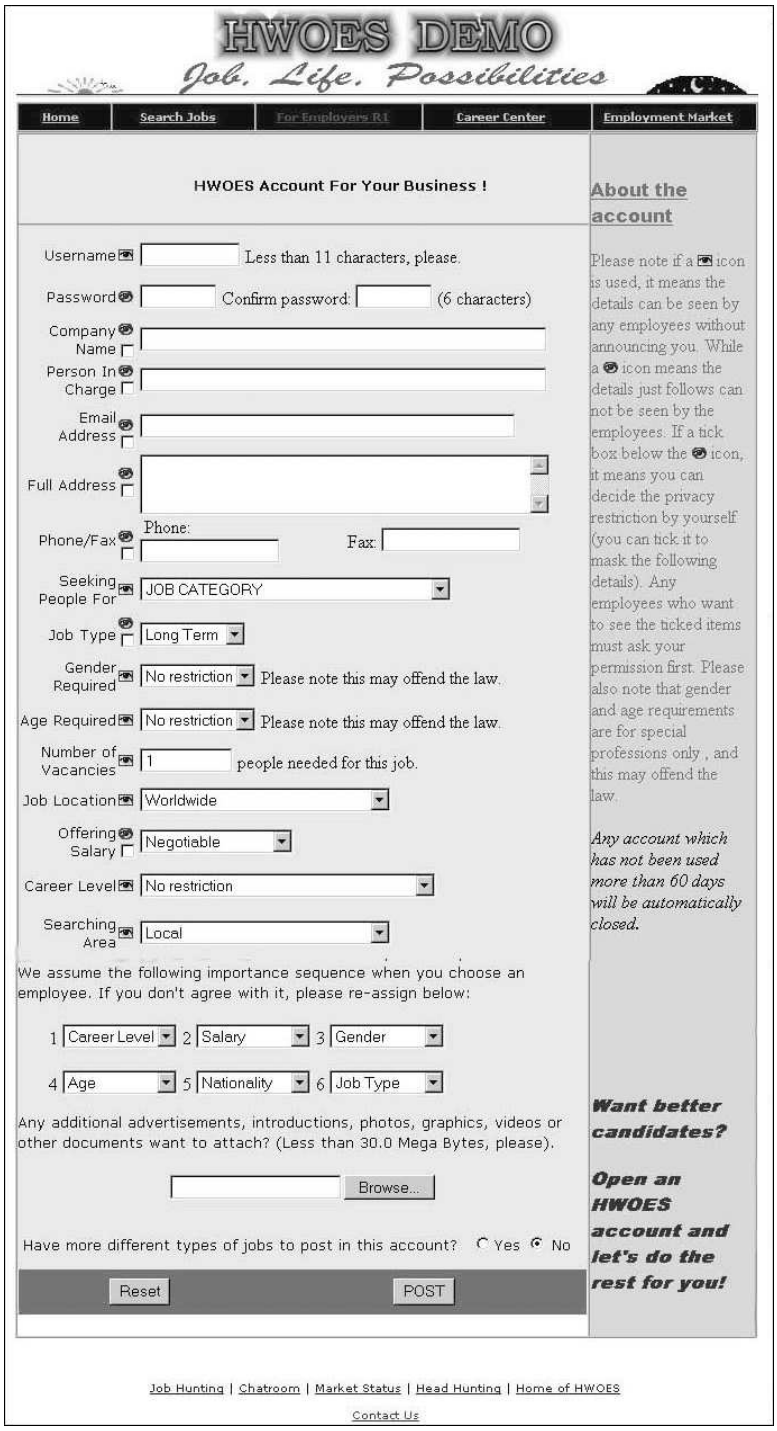

Figure 4: The Sign-in form for Employers [Liu 2001]

HWOES receives information via Forms (see Figure 4) and stores the information in a Personnel Management Space (see Figure 2). This is a Hyperwave database setup especially for HWOES containing personalized accounts and automatically generated attributes [HWTW 1999], which are used for efficient searching. After storing the necessary information about employees and employers HWOES will search through the Personnel Management Space, checking the latest circumstances of the employment market, and return the results to users according to their requirements. Users can then update or delete any of the details they have already sent to HWOES and request new services.

4.1 Functional Components of HWOES

The main functional components of HWOES are illustrated in Figure 3 and run on a Hyperwave Information Server [Maurer 1996]. The Server could be managed by institutional or commercial ISP's. It may be an international server system or just a small regional/local server.

A set of PERL scripts run on the Server and deals with each employment service. The services include establishing and maintaining the database (*PMS*), performing matching processes and serving requests for users.

4.2 Sign-in Forms for Employees and Employers

Sign-in Forms for employers (see Figure 4) and employees provide the necessary information items for HWOES. Key attributes that are used to do the matching are clearly pre-defined in option lists to avoid errors. Text, graphics and even videos (e.g. a inaugural television spot for the client) are also supported as attachment documents. The Form shown in Figure 4 is for demonstration use only. For practical use, some items such as *job category* and *job location* need to be modified and expanded.

Users can set any sensitive items to be visible/hidden (see Section 4.5). The hidden items will only be shown to the account owner. However, an employer, for example, can ask permission from an employee to show relevant hidden details. The completed form will then be sent to the Personnel Management Space (*PMS*, the database) to be stored. Naturally, there is a corresponding Form for employees.

4.3 The Expression of User's Details in a Compact Vector Format

Key attributes for both employers and employees are identical but complementary in meaning (what an employer requires is what an employee should posses, and vice versa). Hence we can use the same *UAPV* format for the two types of users, and the searching process involves matching vector couples.

Attribute Name	**Symbol**	**Pre-defined Values**	**Examples**
Job Category	J	Accounting/Auditing=01 Advertising/Public Relations=02 … … Banking =08 … … Computing=12 … … Engineering=19 Financial Services=20 … …	A computer programmer or job: J12
Job Location	L	Afghanistan=001 Albania=002 Algeria=003 … … New Zealand=111 … … Zimbabwe=174	A job opportunity in New Zealand: L111
Salary	S	<$10,000 =1 <$20,000 =2 <$30,000=3 … … <$60,000=6 … …	Available salary is $52,000: S6

Table 1: An Example of the Pre-Defined Attribute-Value List for UAPVs.

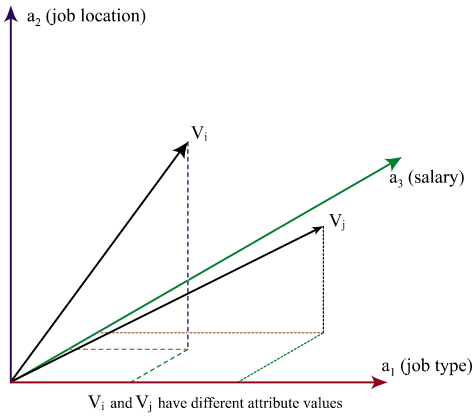

Figure 5: Two 3-Dimensional Vectors Formed from Attributes: location, salary, and job type

Of course *UAPVs* can be extended to as many dimensions as required. The more attributes that are provided, the more accurate *UAPVs* will be when matching.

To keep things concise, yet meaningful, we choose strings that combine characters for attribute names and digits as the corresponding values (actually, index-numbers in pre-defined lists, see Table 1). Using this format, the sequence of each attribute in a *UAPV* can be arbitrarily ordered, and the length of the *UAPV* can be flexible.

For example, see Figure 6, a computer programmer who wants to work in New Zealand with an annual salary above $50,000 has a *UAPV*= (J12L111S6).

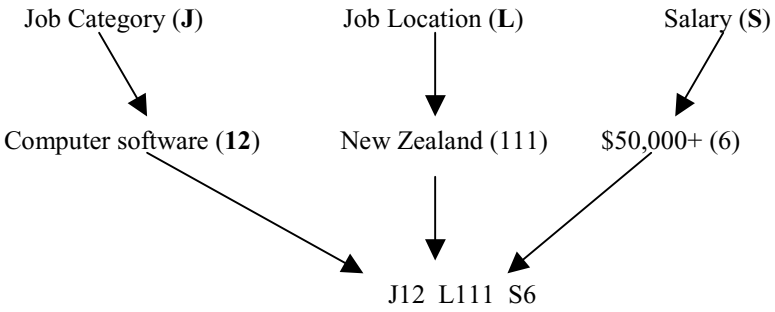

Figure 6: Building a UAPV

Since different users will have different priorities (e.g., some people may regard job location as the most important factor, but others may consider salary is more important) we also need a weighting process to assist *UAPV* matching. The *UAPVs,* together with the importance sequence (weighting), and other system-required information, are then stored in the Hyperwave document management system.

4.4 A Simple *UAPV*-Based Automatic Searching Algorithm

Jobseeker – employer matching can be considered as the problem of finding the most similar complementary vectors in the existing vector space. It is a highly complex science that was not part of our brief. Thus for our prototype we used an extremely simple method based on nearest-neighbor algorithms [Mitchell 1997].

Since the searching task consists of finding the nearest vector in the complementary space, we calculate the standard Euclidean distance between the reference vector and the candidate vectors using formula 1 (see Figure 7).

$$d(V_i, V_{ref}) = \sqrt{\sum_{r=1}^{n} (a_r(V_i) - a_r(V_{ref}))^2} \quad \ldots\ldots\ldots\ldots (1)$$

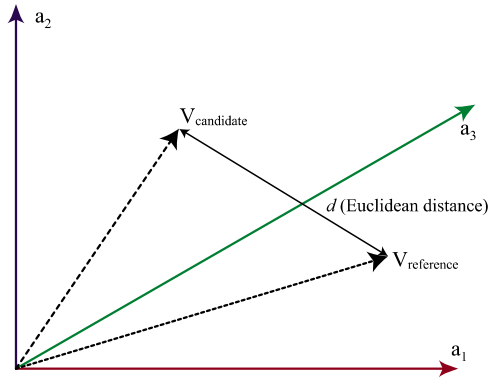

Figure 7: Evaluating the "Distance" From Two Vectors With Different Attribute Values

We then pick the vectors with the smallest Euclidean distances. The employee and employer *UAPVs* are not necessarily continuous along all dimensional axes, however, so the vectors are discrete along some dimensional axes and make the vector space discrete and non-contiguous. This results in two types of attributes: one discrete and the other one evaluated. Another aspect for the Euclidean distance evaluation is weighting. As mentioned previously, not every attribute is necessarily of the same importance for the final result. A customer may feel some attributes are more important than others, so the evaluation formula is altered to:

$$d(V_i, V_{ref}) = \sqrt{\sum_{r=1}^{n} W_r (a_r(V_i) - a_r(V_{ref}))^2} \quad \ldots\ldots\ldots\ldots (2)$$

where *Wr* is the weighting factor for the r-th linear attribute. The system allows flexible weighting factors assigned by users.

For practical use, formula (2) can be simplified into the linear form below:

$$d_i(V_i, V_{ref}) = \sum_{r-1}^{n} W_r \left| a_i(V_i) - a_r(V_{ref}) \right|$$

$$= w_1 \left| a_{i1} - a_{ref1} \right| + w_2 \left| a_{i2} - a_{ref2} \right| + w_3 \left| a_{i3} - a_{ref3} \right| + w_4 \left| a_{i4} - a_{ref4} \right| + \ldots + w_n \left| a_{in} - a_{refn} \right| \quad \ldots\ldots\ldots\ldots\ldots\ldots (3)$$

where d_i is the distance value of the instance i being searched in the database.

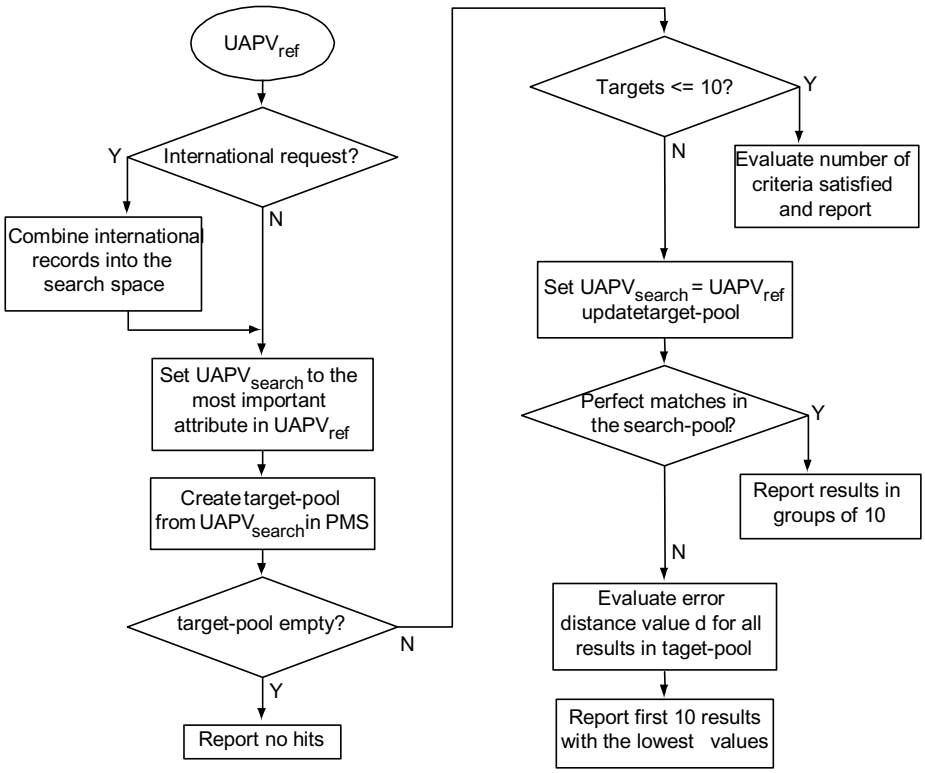

Figure 8: Flowchart for Calculation of Search Results in HWOES

On the Hyperwave server *UAPVs* are actually stored as an additional object attribute (an indexed *keyword*) for each document object belonging to a particular user. Hyperwave supports arbitrary numbers of indexed attributes that can be added to an object. The new attribute at issue is called a UserPattern.[1]

[1] It is the same as the UAPV, but with a different name to avoid confusion with the object attributes in the Hyperwave system.

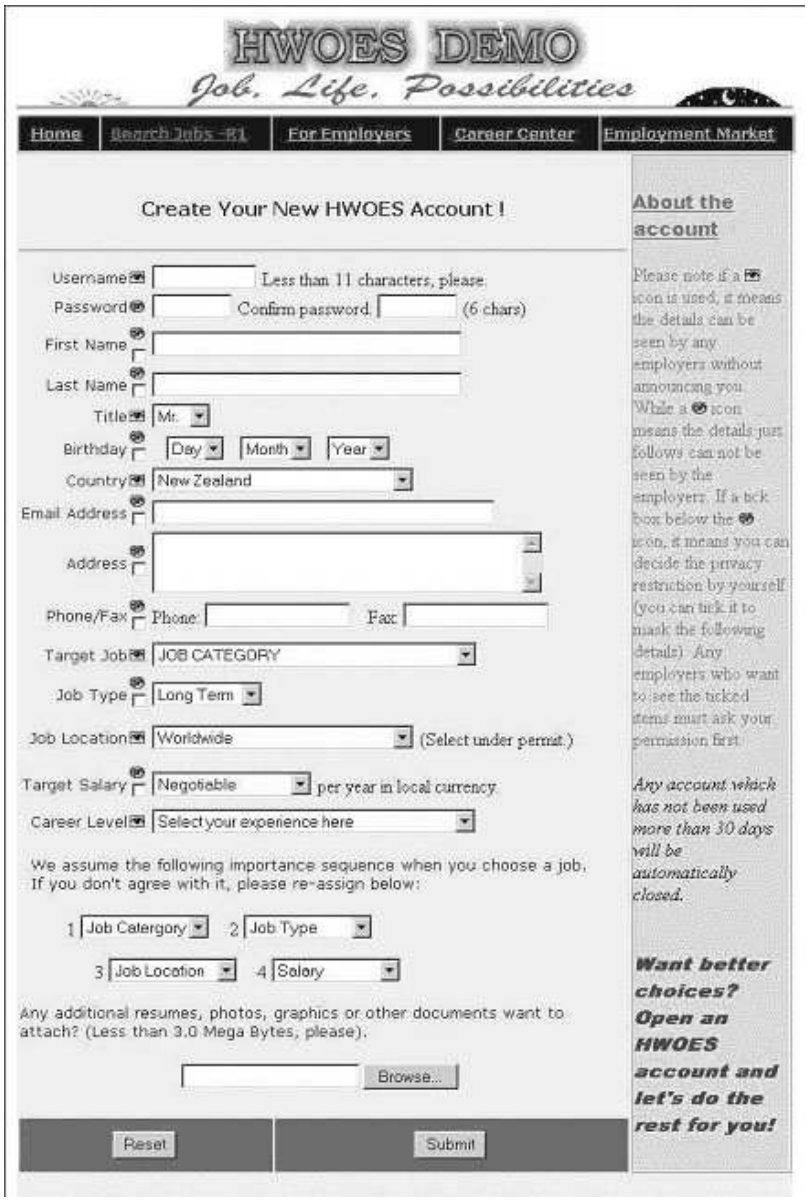

Figure 9 The Information form for Employees [Liu 2001]

4.5 Levels of Privacy in HWOES

As elaborated in Section 2 users should be able to determine when (if ever) their private details are made visible, and HWOES does just this. Figure 9 shows the Form for a jobseeker to enter their personal details. In the Form, the eye icon indicates that a prospective employer can see an item, whereas a negated eye indicates that,

at present, the data is restricted. In the screen shot the icons indicate the default settings for the data. Of course, these settings will be changed later in the negotiation process.

User notes (not shown in the figure), explain the icons to the user. A check box below the icon allows the user to change the state from visible to hidden and conversely. Any user who wants to see the ticked items must ask the owner for permission first.

4.6 Iterative Interactions Between Jobseekers and Employers

Figure 10 is an example of the negotiations when an employer first logs in. Each account consists of several basic items:

- The posted sign-in Form details (displayed here as the Account Form link).
- The attached documents (e.g., advertisement pictures).
- System-generated messages (e.g., answers to job advertisements).
- A system generated new! alert icon to indicate new matches (this generates a new Form for further interaction).

There is a corresponding page for job seekers.

Figure 10: An Example of a Successful Member Interaction (Employer) [Liu 2001]

5 International services

Unlike the simple Web links to remote servers that many current OESs use for their international services, worldwide collaboration between separate HWOESs can be achieved in a more active and effective way. Collaborative HWOESs need only a small amount of information exchanged (*UAPVs and GOids (Global Object id*[2]*)* couples) for the searching process to be performed with a worldwide scope. The *UAPV* is tiny in size but it delivers all necessary details needed for searching; and the *GOid* gives the full address where the original *UAPV* is actually stored. HWOES can use a broadcasting algorithm to exchange *UAPV* and *GOid* couples from geographically separated systems. Full target details are only retrieved from remote sites when a match is found and final reports to customers are generated.

6 Conclusion

The system prototype described gives users significant control over the levels of privacy they need. Most significantly, virtual connections supporting negotiation between jobseekers and employers are set up in real time and can be modified in real time. Finally, an efficient worldwide service can be implemented using the small UAPVs and GOids.

References

[Borghoff and Pareschi 1997] Borghoff U., Pareschi R. (1997). Information Technology for Knowledge Management; *J.UCS* 3, 8 (1997), 835-842.

[CntPub 2000] How Online Recruitment Is Shaking Up Its Own Industry, New Media Age, Centaur Publishing Ltd. (April 13, 2000), 41.

[HeadH 2001] The HeadHunter Company - http://www.headhunter.net/.
[HotJ 2001] The HotJobs Company - http://www.hotjobs.com.

[HW 2001] The Hyperwave portal http://www.hyperwave.com.

[HWTW 1999] The Technical White Paper of Hyperwave Information Server V5.0, 1999, http://www.hyperwave.com.

[Maurer 1996] Maurer H. Hyperwave: The Next Generation Web Solution. Addison-Wesley (1996), UK.

[Jobs 2001] The Jobs Company - http://www.jobs.com.

[Liu 2001] Liu, H. Hyperwave Online Employment System, University of Auckland, Thesis (March 2001), New Zealand.

[2] Every object stored in a Hyperwave server has a unique global id, consisting of a unique server id and object id.

[Mnstr 2001] The Monster Company - http://www.monster.com

[Mitchell 1997] Mitchell, T. M. Machine Learning, McGraw-Hill (1997), USA.